D E S C R I P T I O N

Method and System for Document or Content Off-Loading to a Document Repository

5

Background of the Invention

The invention relates to data processing environments with large document repositories and more specifically to a method and system
10 for off-loading a document's content from a document processing system to a remote repository.

Known client mailing applications like Lotus™ Notes™ or Microsoft™ Outlook™ contain continuously growing document repositories, namely
15 the incoming and outgoing notes or emails often including large attachments like text documents, graphics or even storage consuming digitized pictures. Therefore, e.g., a Lotus Notes application uses a Lotus Domino™ database from which a tool like IBM Content Manager CommonStore™ for Lotus Domino (CSLD) is used to move documents
20 stored in that database to an archive physically located on a different device like a tape storage. CSLD thereupon allows to access documents that have previously been archived.

CSLD also allows to access documents that have been archived from
25 any archive client application (e.g., scanning applications, CommonStore for SAP™, etc). When documents are retrieved from the archive to a Notes database, a Lotus Notes document is created.

In most scenarios, such documents are viewed only once with a Notes
30 internal or external viewer, and then become obsolete. However, such temporary retrieval documents waste resources and have impact on the overall performance of the Notes application. Therefore, users have to delete these documents. But since the main interest of a user is to view an archived document, there is actually no

need to retrieve a document to Lotus Notes.

Summary of the Invention

It is therefore an object of the present invention to provide a
5 method and system for handling content off-loading from a document
processing system to a large repository which is less resource
consuming than the prior art approaches.

Another object is to provide such a method and system which allow
10 to retrieve off-loaded content, minimally wasting resources.

It is yet another object to provide such a method and system which
enables viewing of off-loaded content in a user-friendly way.

15 The above objects are achieved by the features of the independent
claims. Advantageous embodiments are subject matter of the
subclaims.

The idea underlying the invention is to provide a URL link to
20 off-loaded content and to enable to display the content in a
viewing application. In particular, it is proposed to detach the
content from a document, to transfer it to a remote repository, and
to replace it by a placeholder text implemented as a URL link. The
text can contain information, e.g., about who off-loaded or
25 archived the document/content, the time/day of off-loading, and the
original attachment filename, in order to identify the off-loaded
content. The URL link, for instance, is a Notes URL link hotspot
richtext element. That is, when clicked, a browser is opened,
displaying the content associated with the URL link.
30

That solution is less resource consuming than the prior art
approaches, particularly regarding storage capacity and network
traffic. It can advantageously be used, e.g., in mail clients where
mail documents, content or attachments are archived to a remote

repository server, and can be viewed directly without physically transferring them to the mail client.

It is understood hereby that the above mentioned remote repository
5 server can also be a local hard disk.

The invention can be applied to every known mail client program or system and anables worldwide viewing of a document. Preferably, archived documents can be viewed within the mail client via the URL
10 links, e.g., by using a common web browser either as a plug-in to the mail client or a separate web browser that is automatically started when the URL is clicked.

Preferably, an underlying mail server, e.g. Domino server, is
15 connected to a web dispatcher component, which is basically a stripped-down web server with special archive-related functionality. The web dispatcher provides web access to an archived content. Hereby, requests to be processed by the web dispatcher are sent as HTTP requests with a defined parameter set.
20

In another embodiment, when a document is off-loaded, it is assigned a unique identifier (ID). The ID becomes part of the URL and can be encrypted for security reasons.

25 In another embodiment, the document's type or the document's content type is stored with a document when the document is off-loaded. Hereby, the aforementioned web dispatcher can maintain a mapping table mapping content types to MIME types. This allows the browser to interpret each file correctly.
30

In yet another embodiment, the aforementioned browser viewing is performed from within a search hitlist, i.e. when a search over the repository returns a hitlist document. For every hit in the hitlist, a button and an URL link hotspot are displayed. When the

button is pushed, the corresponding content is retrieved. When the URL link is pushed, the content is viewed in a plug-in or separate web browser. This allows to quickly view content to find out whether it is the desired one. Then, if necessary, it can be

5 retrieved back to the mail client.

It is noteworthy that, instead of using a web browser for viewing an off-loaded content, every kind of HTTP client tool can be used.

Brief Description of the Drawings

In the following, the present invention is described in more detail
5 by way of embodiments from which further features and advantages of
the invention become evident whereby

Fig. 1   is an overview block diagram showing a document before
             and after content off-load according to the invention;
10
Fig. 2   is a flow diagram illustrating basic components and data
             flow of a preferred embodiment of the invention;

Fig. 3A   is a diagram showing various steps of a content off- load
15   procedure according to the invention;

Fig. 3B   is another diagram showing various steps of content
             retrieval according to the invention; and

20 Fig. 3C   is another diagram another embodiment of content
             retrieval via a search over a repository.

Detailed Description of the Drawings

Fig. 1 illustrates the basic concept of the invention by showing a
5 document before and after content off-load according to the
invention. A mail client 101 that has stored a number of email
documents 102 - 104 (doc1, ...), each of them containing content
105 (XYZ), and possibly one or more attachments 106.

10 During an off-loading procedure, the content 105 and the possible
attachments 106 are detached from the document 102 and transferred
107 to a remote repository server 108. In the original document
102, after the off-loading 107, the content 105 is replaced by a
placeholder text 109, i.e. a Lotus Notes URL link hotspot richtext
15 element in the present example. Possible attachments 106 are
replaced by a corresponding URL.

The block diagram depicted in Fig. 2 shows a Lotus Notes
environment as an example of a document processing system. The
20 system is shown in a state after an already performed off-loading
procedure. It comprises a Notes database 201 (Notes DB) for which
an exemplary eMail document 202, where the document content were
replaced by a URL link hotspot richtext element 203 as discussed
beforehand. The URL text contains information about who archived
25 the document, time/day of archiving, and the original attachment
filename. For example

<u>**<<< Attachment 'CSLDClient.sys' has been archived by user**</u>
<u>**'Daniel Haenle/Germany/IBM' on '09/20/2000 11:32:18 AM'. >>>**</u>

30
When the URL link 203 is clicked, a browser 204 (in the present
example a Netscape™ web browser) is started, connecting to the
given URL 203. When a document is off-loaded by CSLD 206, it is
assigned a unique identifier (ID). The ID is encrypted for security

reasons and becomes part of the URL 203. An HTTP „GET" request together with the ID is sent from the web browser to an HTTP dispatcher 205 which is a stripped-down HTTP server. The goal of the HTTP dispatcher 205 is to provide web access to archive

5 content. Requests to be processed by the HTTP dispatcher 205 are sent as HTTP requests with a defined parameter set.

The CSLD HTTP dispatcher 205 extracts the encrypted ID from the URL 203, decrypts it, retrieves the content from the repository, and

10 sends it to the browser, where it is displayed. Of course, for some document types a special browser plugin is required.

The HTTP dispatcher 205 forwards the request to IBM Content Manager CommonStore™ for Lotus Domino 206 (CSLD) and requests the content

15 having the sent ID, referred to in the URL 203. The CSLD 206, in particular, provides an interface to one or more document repositories 207 - 209. The repository or repositories, in the present embodiment, is (are) comprised of Tivoli™ Storage Manager ™ 207 (TSM), Content Manager™ 208 and Content Manager™ OnDemand™

20 209. Each of these three components 207 - 209 can be connected to one or more tape storage devices 210 - 212. TSM 207 retrieves the content requested by the HTTP "GET" request and returns it to CSLD 206. Finally, the retrieved content can be viewed using the Netscape browser 204.

25

A complete URL 203 computed by CSLD 206 during off-loading consists of the IP address or host name running the HTTP dispatcher 205, the HTTP dispatcher port, the internal command *sGet* and an encrypted document ID. An example is

30

http://popken.boeblingen.de.ibm.com:8085/?sGet&Dl1eTH1W
Xw1jABdcAIF5XBJqYn8HCHRhIX9nC2VmYXd%2Ba1J
XAEJ5XBJXTkRRa0FuCEBDUUBdEgAAeQRtMjA4LzM
VMDBNMTgM

35

When a document is off-loaded by CSLD 206, the document's content type is stored with the document. The HTTP dispatcher 205 maintains a (not shown) table mapping content types to MIME types. This allows the web browser 204 to interpret the file correctly.

5

It is noted that the browser viewing feature has nothing to do with Notes except that the URL link 203 is kept in a mail document 202. Therefore, no temporary Notes retrieval documents are created.

10 Fig. 3A is a diagram showing various steps of a content off-load procedure according to the invention illustrated for attachment archiving. A user starts 301 the off-load procedure by, e.g., pushing an 'archive' button in the Notes client. Alternatively, the procedure can be triggered automatically 303. The attachment is
15 detached 302 by CSLD and moved 304 to a repository. Afterwards, CSLD replaces 305 the attachment(s) by a URL link.

Fig. 3B shows the scenario for a single content retrieval in case of an off-loaded attachment. The user initiates 311 retrieval by
20 clicking the URL link which opens 312 a web browser. The web browser sends 313 an HTTP "GET" request to the server designated in the URL. The HTTP server retrieves 314 the attachment from the repository via CSLD. The content is sent back 315 as an HTTP response to the web browser. Finally, the browser displays 316 the
25 attachment.

Fig. 3C shows the scenario for retrieving an attachment via a search over the repository. A user initiates 321 a search in the repository. CSLD performs 322 that search and returns the result as
30 a Notes hitlist document. From that hitlist, the user can click 323 on a URL representing a certain hit. This opens 324 a web browser. The web browser sends 325 an HTTP "GET" request to the server specified in the URL. The HTTP server retrieves 326 the attachment from the repository via CSLD. The content is sent back 327 as an

HTTP response to the web browser. Finally, the browser displays 328 the attachment.

It should be noted that the above described browser viewing shall
5 not be confused with the browser viewing feature in a Domino web client. With a Notes web client, Notes databases are accessed from within a browser. With CSLD browser viewing, content in an archive is viewed in a browser without retrieving the content to Lotus Notes. Browser viewing also works with the Notes web client. That
10 is, it makes no difference whether a document URL link is clicked in a document being viewed in the Notes client or in a document being viewed in a Domino web client. In both cases, no Lotus Notes document is created.

15 CSLD browser viewing also allows users to forward an URL link to other users, even to those who have no Notes client installed. All these users will be able to view the document in a browser. A further application of CSLD browser viewing is viewing of archived documents for which no Notes viewer exists, but which are supported
20 by a browser (native or via plugin).

25